

6.S092

Lecture 3

Reminders

Assignment 1 is due

TONIGHT

11:50PM

Last Time

- PDTs, PDSs
- COnTrol Flow and Loops
- Privacy
- Methods and Modularity
- Functions called Methods in JAVA
- Function signatures

Today

- The switch statement
- Object
- Class - Abstract Class
- Inheritance
- Enumerated Values
- Interfaces (DID NOT COVER)
- Recursion (May be) (DID NOT COVER)

Switch Statement

Like if/else if/else structure

Based on “cases”

Keywords to remember: `switch`, `case`,
`break`, `continue`, `default`

Object Oriented Programming OOP

Creating an abstract concept of a “thing”

Picture a car.

Did you think of a car or a SPECIFIC type of car?

What is a car?

Google's definition: a road vehicle, typically with four wheels, powered by an internal combustion engine and able to carry a small number of people.

So a car is ...

A vehicle (Description)

has number of wheels (n, Instance variable)

has an engine (n, Instance variable)

moves people (v, Instance method)

In Java how do you make a “Car”?

This is what “class” is.

A class in Java is usually the BLUEPRINT for the thing it codes.

public class Car

```
public class Car {  
    //The stuff in here is Java's  
    // representation of a car  
}
```

Properties of a car

- They all have wheels (could be of different amount)
- They all have windows (could be of different amount)
- They all have an engine (could be of a different type)

How do we then represent a car?

With instance variables.

These are variables declared after the open braces and in that scope.

Back to the code

```
public class Car {  
    private int numWheels;  
    private int numWindows;  
    private EngineType engine;  
  
    //OTHER STUFF  
}
```

Let's look at ComplexNumber again

